Manuel d'utilisation du Pretty Printer

L'execution se fait en mode console par la commande :

./pp in [out] [--]

- in : Il est obligatoire de fournir le fichier d'entrée.

- [out] *optionnel* : Par défaut le fichier de sortie est le STDOUT (soit la fenêtre de la console), on peut spécifier ici un nom de fichier de sortie.

- [--] *optionnel* : Si on rajoute le signe "--" le Pretty Printer supprimera tous les whitespace superflus à l'exception de ceux écrit en mode commentaire, en mode caractère, entre guillemets ou lorsqu'un # a été lu en début de ligne.

/------------------------/

Détails du fonctionnement

Le programme est à la base une implémentation informatique d'un automate. L'état de celui-ci évolue selon les 10 champs donnés ci-dessous :

{

nb_ind	:	nombre d'accolades ouvertes non fermées
was_CR	:	booléen indiquant si la dernière chose significative (pas espace ou tab) était un retour à la ligne: '\n' ou '\r'
was_slh	:	booléen indiquant si le dernier char lu est /
was_bslh	:	booléen indiquant si le dernier char lu est \
nb_qm	:	nombre de ? vus consécutivement (pour les trigraphes en norme ANSI)
qte_on	:	booléen indiquant si on est dans des guillemets
pnd_on	:	booléen indiquant si il y a un # en début de ligne, i.e. si on est dans une def de macro
car_on	:	booléen indiquant si on est dans une def de caractère (entre deux ')
com_on	:	booléen indiquant si on est en mode commentaire
was_star	:	booléen indiquant si le dernier char lu est * (pour fermer les commentaires)

}

Les fonctions du programme représentent les transitions possibles sur cet automate. Ces transitions sont à leur tour appelé par la fonction "main" selon la nature des caractères lus dans le fichier et l'état courant de l'automate. Cette fonction effectue aussi quelques actions de formatage comme la gestion des espaces en début (et éventuellement en cours de ligne.

Elles sont nommées suivant l'état de l'automate qu'elles traitent avec la nomenclature :
1 si le champ contient la valeur vraie
0 si la valeur est fausse
x pour signifier « don't care »

Voici le détail des fonctions de transition :

- tr_default : C'est la fonction appellé si l'automate est dans un état "standard". Il n'y a pas de commentaires ouvert, de guillemets, nous ne venons pas de passer à la ligne, etc. Selon le caractère lu elle oriente l'état de l'automate vers des cas specifiques.

- tr_n_x_x_x_m_x_x_x_1_0 : Nous sommes en commentaire normal et le dernier caractère lu n'était pas une étoile ("*"). Les accolades ("{" et "}") n'ont aucun effet particulier. Les passages à la ligne provoquent une indentation mais pas une sortie du mode commentaire.
Toutefois le commentaire est fermée à la fin de la ligne et reouvert en début de la suivante.

- tr_n_x_x_x_m_x_x_x_1_1 : Nous sommes en commentaire normal et le dernier caractère lu était une étoile ("*"). Soit le commentaire est poursuivi, soit en cas de slash ("/") nous sortons du mode commentaire.

- tr_n_x_x_x_m_x_x_1_x_x : Nous sommes en mode dièse ("#"). On attend simplement le prochain retour à la ligne. Les caractères spéciaux n'ont pas d'effets particuliers, sauf le cas des trigraphes.

- tr_n_x_x_x_m_x_1_x_x_x : Nous sommes en mode caractère. On attend la fermeture du caractère. Les caractères spéciaux n'ont pas d'effets particuliers, sauf le cas des trigraphes.

- tr_n_x_x_0_m_1_x_x_x_x : Nous sommes avec des guillements ouverts et le dernier caractère lu n'est pas un backslash ("\"). On attend la fermeture des guillemets. Les caractères spéciaux n'ont pas d'effets d'effets particuliers, sauf le cas des trigraphes.

- tr_n_x_x_1_m_1_x_x_x_x : Nous sommes avec des guillements ouverts et le dernier caractère lu est un backslash ("\"). Soit c'est un caractère d'échappement pour un \" qui ne ferme pas les guillemets, soit on continue normalement. On attend la fermeture des guillemets. Les caractères spéciaux n'ont pas d'effets d'effets particuliers, sauf le cas des trigraphes.

- tr_n_0_x_x_2_x_x_x_x_x : Nous ne sommes pas en début de ligne et nous venons de lire "??", soit un début de trigraphe éventuel. Il n'y a pas de commentaire, guillemet ou caractère actif. S'il y a lecture de "=" avec "??=" correspondant à un dièse nous ne passons pas en mode dièse.

- tr_n_1_x_x_2_x_x_x_x_x : Nous sommes en début de ligne et nous venons de lire "??", soit un début de trigraphe éventuel. Il n'y a pas de commentaire, guillemet ou caractère actif. S'il y a lecture de "=" avec "??=" correspondant à un dièse nous passons en mode dièse.

- tr_n_1_x_x_m_x_x_x_x_x : Nous venons de passer à la ligne et nous avons lu strictement moins de 2 "?". Il n'y a pas de commentaire, guillemet ou caractère actif.

- tr_n_0_1_x_m_x_x_x_x_x : Nous ne venons pas de passer à la ligne et nous avons lu un slash ("/"). Il y a un passage possible en mode commentaire.

- tr_n_1_1_x_m_x_x_x_x_x : Nous venons de passer à la ligne et nous avons lu un slash ("/"). Il y a un passage possible en mode commentaire.

- tr_n_1_1_x_m_x_x_x_x_x : Nous venons de passer à la ligne et le premier caractère qui y a été lu est un slash ("/"). Ceci rend possible l'entrée en commentaire normal ("/*") ou en commentaire local (//). Ce dernier est comparable à une dièse ("#") en début de ligne.

Remarques générales :

Les espaces ou tabulations fournis par l'utilisateur en début de chaque ligne sont automatiquement supprimés et remplacés par l'indentation automatique du Pretty Printer. Cette indentation automatique s'effectue toujours après la lecture du premier caractère et avant la première écriture dans le fichier correspondant à la nouvelle ligne. Ceci afin de tenir compte des éventuels trigraphes et commentaires.

Dans le code source les variables suivantes sont couramment utilisées :
- IN, le canal de lecture sur le fichier d'entrée.
- OUT, le canal d'écriture sur le fichier de sortie.
- state, l'automate.
- zapper, booléen valant 0 par défaut et 1 si l'option [-] a été fourni par l'utilisateur.

La valeur de retour des fonctions de transition DO / DONT signale à la fonction main s'il doit effacer les prochains espaces fournis par le fichier d'entrée ou non.

La constante DEBUG défini en début du fichier de code source permet d'activer le mode de débogage en ainsi d'afficher les transitions et les états intermédiaires de l'automate au cours du traitement d'un fichier. Pour l'activer sa valeur doit être mise à 1 avant compilation, 0 sinon.